

TITLE OF THE INVENTION  
IMAGE PROCESSING APPARATUS AND METHOD THEREOF

BACKGROUND OF THE INVENTION

5   Field of the Invention

        The present invention relates to an image processing apparatus and method thereof and, more particularly, to an image process that realizes preferred color reproduction at high speed.

10

        Background of the Invention

        An image processing apparatus which makes color correction often uses  $3 \times 3$  matrix operations and gamma conversion so as to attain a high-speed color process and a reduction of the circuit scale. Such matrix operations and gamma conversion can achieve arithmetically faithful color conversion, but cannot achieve preferred color reproduction in correspondence with one's memory colors and favor.

20

        In order to convert color-corrected data into a CMYK signal depending on an output device, the color corrected data undergoes luminance-density conversion, and then undergoes a masking process, under color removal process, and black correction process. These processes impose heavy loads, and it is difficult to attain a size reduction of that circuit.

25

        As a method that can compensate for the above

drawbacks, a color difference diffusion (CD) method disclosed in Japanese Patent Laid-Open No. 11-55535 is known. With this method, the distances on a color space are calculated for respective combinations of colors that an output device can reproduce, so as to associate colors closest to those of input image data with each other, and a combination of colors that can minimize the distances is selected. Furthermore, this method is a technique which can implement an arrangement for diffusing the differences among associated colors to other pixels by a small-scale circuit, and can improve a high-speed process and color reproducibility.

However, even when the CD method which allows high-speed color separation is used, it is difficult to attain preferred color reproduction corresponding to various user's requirements using a small-scale circuit.

## SUMMARY OF THE INVENTION

The present invention has been made to solve the aforementioned problems individually or together, and has as its object to improve color reproducibility.

In order to achieve the above object, a preferred embodiment of the present invention discloses an image processing apparatus, comprising: a converter, arranged to color-convert input image data using a

three-dimensional table selected from a plurality of three-dimensional tables and an interpolation process; a first calculator, arranged to obtain error-corrected data by adding error data to the color-converted image data; an output section, arranged to select a dot pattern from a combination of dot patterns selected from a plurality of combinations of dot patterns on the basis of the error-corrected data, and output the selected dot pattern; and a second calculator, arranged to obtain error data by calculating a difference between a predetermined value corresponding to the output dot pattern, and the error-corrected data.

Other features and advantages of the present invention will be apparent from the following description taken in conjunction with the accompanying drawings, in which like reference characters designate the same or similar parts throughout the figures thereof.

#### BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram showing the arrangement of an image processing apparatus according to an embodiment of the present invention;

Fig. 2 is a block diagram showing the detailed arrangement of the image processing apparatus shown in Fig. 1;

Fig. 3 is a block diagram showing the detailed

arrangement of a color converter shown in Fig. 1;

Fig. 4 shows an example of a three-dimensional lookup table (3D-LUT) shown in Fig. 3;

Fig. 5 is a view for explaining the concept of an  
5 interpolation unit shown in Fig. 3;

Fig. 6 is a block diagram showing the detailed arrangement of a quantization processor shown in Fig. 1;

Fig. 7 is a flow chart showing a process to be  
10 executed by the quantization processor;

Fig. 8 shows an example of color difference diffusion coefficients;

Fig. 9 shows an example of an output color table;

Fig. 10 is a flow chart showing a process for  
15 diffusing color differences generated in a quantization process;

Fig. 11 shows an example of an output density table;

Fig. 12 shows an example of color difference  
20 diffusion coefficients;

Figs. 13 and 14 are views for explaining determination of an output color table; and

Figs. 15 and 16 show examples of tables.

## 25 DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

An image processing apparatus according to an embodiment of the present invention will be described

in detail hereinafter with reference to the accompanying drawings.

[Arrangement]

5           Fig. 1 is a block diagram showing the arrangement of an image processing apparatus of this embodiment.

Referring to Fig. 1, an input section 1 inputs multi-valued luminance color data. Note that the input section 1 may incorporate a circuit for correcting  
10 colors and tone characteristics depending on an input system. The input section 1 is not limited to a reading device such as a scanner or the like, and may input RGB luminance data obtained by converting a color image which is received by a facsimile apparatus having  
15 an interface with a medium (a memory card or the like) that stores images, communication unit, and MODEM unit via a known facsimile communication.

A color converter 2 comprises a table memory, interpolation arithmetic processing circuit, and the  
20 like, which are represented by techniques disclosed in, e.g., Japanese Patent Laid-Open No. 8-237497 and the like. The detailed arrangement of the color converter 2 will be described later.

A quantization processor 3 comprises a circuit  
25 for executing various image processes such as a variable magnification process, gamma correction, noise filtering, edge emphasis, and the like, a quantization

circuit, and the like. The arrangement of the quantization circuit will be described later. The quantization processor 3 uses the aforementioned CD method.

5           A print section 4 comprises a printer represented by an ink-jet printer or laser beam printer, and prints a dot pattern output from a memory using four colors, i.e., C, M, Y, and K.

Fig. 2 is a block diagram showing the detailed  
10 arrangement of the image processing apparatus shown in Fig. 1.

Referring to Fig. 2, a CPU 11 controls various functions of the image processing apparatus, and executes an image processing program stored in a ROM 16  
15 in accordance with predetermined operations at a console 15.

A scanning unit 14 which comprises a charge coupled device (CCD) corresponds to the input section 1 shown in Fig. 1. That is, the scanning unit 14 reads a  
20 document image and outputs red (R), green (G), and blue (B) analog luminance data. Note that the scanning unit 14 may comprise a contact image sensor (CS) in place of the CCD.

Also, a card interface 22 corresponds to the  
25 input section 1 shown in Fig. 1, and reads image data, which is sensed by, e.g., a digital still camera (DSC) and is recorded on a memory card or the like, in

accordance with a predetermined operation at the console 15. Note that the color space of the image data read via the card interface is converted from that (e.g., YCbCr) of the DSC into a standard RGB color space (e.g., NTSC-RGB or sRGB) by an image processing unit 12, if required. The read image data undergoes various processes such as resolution conversion into an effective number of pixels and the like required for an application on the basis of its header information, as needed.

The image processing unit 12 corresponds to the color converter 2, quantization processor 3, and the like shown in Fig. 1. The image processing unit 12 executes an image process to be described later, and quantized color data obtained by that image process is stored in a RAM 17. When the quantized color data stored in the RAM 17 reaches a predetermined size required for a print process of a print unit 13 corresponding to the print section 4 in Fig. 1, the print unit 13 prints an image corresponding to the quantized color data on a recording medium.

A nonvolatile RAM 18 comprises, e.g., an SRAM or the like which is backed up by a battery, and stores data unique to the image processing apparatus and the like. The console 15 comprises a photo direct print start key used to select image data stored in a storage medium and to start a print process, a copy start key

in a monochrome or color copy mode, mode keys used to designate modes such as a copy resolution, image quality, and the like, a stop key used to stop a copy operation and the like, a ten-key pad used to input a copy count, a register key, and the like. The CPU 11 detects the depression states of these keys, and controls the respective units in accordance with the detected states.

A display unit 19 comprises a dot-matrix type liquid crystal display (LCD) and an LCD driver, and makes various displays under the control of the CPU 11. Also, the display unit 19 displays thumbnails of image data recorded in a storage medium. The print unit 13 comprises an ink-jet head of an ink-jet system, a versatile IC, and the like. The print unit 13 reads out print data stored in the RAM 17 and prints it out as a hard copy.

A drive unit 21 comprises stepping motors for driving paper feed and discharge rollers in operations of the aforementioned scanning unit 14 and print unit 13, gears for transmitting the driving forces of the stepping motors, driver circuits for controlling the stepping motors, and the like.

A sensor unit 20 comprises a print sheet width sensor, print sheet sensor, document width sensor, document sensor, print medium sensor, and the like. The CPU 11 detects the states of a document and print



sheet on the basis of information obtained from these sensors.

[Color Converter]

5           Fig. 3 is a block diagram showing the arrangement of the color converter 2.

          A three-dimensional lookup table (3D-LUT) 31 outputs data corresponding to the upper 4 bits (total of 12 bits) of each of input luminance data Rin, Gin, and Bin. An interpolation unit 32 makes an  
10       interpolation operation by receiving data Ro, Go, and Bo output from the 3D-LUT 31, and the lower 4 bits (total of 12 bits) of each of input luminance data Rin, Gin, and Bin, and outputs output luminance data Rout, Gout, and Bout (8 bits each). With this interpolation  
15       operation, the table size of the 3D-LUT 31 can be reduced. Note that the number of grid points of the 3D-LUT 31 is determined on the basis of the balance between the quality of an image to be processed and the  
20       characteristics on the output side, and the table size.

          Fig. 4 shows an example of the 3D-LUT 31. A case will be exemplified below wherein the number of grid points is 17, and the spacing between neighboring grid points is 16.

25           A table shown in Fig. 4 is generated to represent combinations between Rin, Gin, and Bin, and Ro, Go, and Bo using input addresses, which are obtained by

extracting the upper 4 bits of input luminance data  $R_{in}$ ,  $G_{in}$ , and  $B_{in}$ , and combining the extracted bits. In this case, this table records 4,095 combinations.

If input luminance data  $R_{in}$ ,  $G_{in}$ , and  $B_{in}$  correspond to a grid point, output data  $R_o$ ,  $G_o$ , and  $B_o$  can be directly obtained from the table. However, it is not practical to prepare for a table that describes input/output relationships for all inputs, e.g.,  $256^3 = 16,770,000$  colors (each of R, G, and B data = 8 bits). Hence, output data other than the grid points (representative points) of the table are calculated by the following interpolation process by the interpolation unit 32.

When input luminance data are applied to a space (hexahedron) divided by  $17 \times 17 \times 17$  grids, a relationship between the point (input luminance data) included in the hexahedron and surrounding eight grid points is determined. Output luminance data  $R_{out}$ ,  $G_{out}$ , and  $B_{out}$  can be calculated from output data  $R_{oi}$ ,  $G_{oi}$ , and  $B_{oi}$  ( $i = 1$  to  $7$ ) corresponding to these eight grid points. However, when the point (input luminance data) included in the hexahedron is defined by the eight grid points, complicated calculations are required. Hence, the space (hexahedron) is further segmented using tetrahedrons each of which couples four grid points.

The segmentation method is implemented as follows

by calculating the differences from a reference point. In this embodiment, an explanation will be given using a two-dimensional (2D) space shown in Fig. 5 for the sake of simplicity. Let  $(x, y) = (16, 16)$  be the  
5 reference point of grids, and  $(x_{in}, y_{in}) = (28, 20)$  be the input.

Differences  $\Delta x = x_{in} - x$  and  $\Delta y = y_{in} - y$  of the input from the reference point are compared. If  $\Delta x > \Delta y$ , it is determined that the input belongs to region  
10 A shown in Fig. 5. Likewise, in the 3D space, the differences of the input from the reference point can be calculated and one of six tetrahedrons to which the input belongs can be determined. After a tetrahedron to which the input belongs is determined, output data  
15  $R_o$ ,  $G_o$ , and  $B_o$  corresponding to the input can be calculated from output data  $R_o$ ,  $G_o$ , and  $B_o$  of four grid points corresponding to that tetrahedron by linear interpolation.

Note that conversion from one luminance space  
20 into another luminance space has been explained. If density data (or data of another color space) must be input to the next quantization processor 3, luminance-density conversion or conversion into another color space can be combined with the aforementioned  
25 conversion. In this case, if more grid points are used, errors due to nonlinear conversion can be reduced, and the conversion precision can be improved.

The grid spacing may be constant. However, nonuniform grid spacings may be adopted to obtain a denser grid spacing in a region that requires conversion with higher precision.

5

#### [Quantization Processor]

Fig. 6 is a block diagram showing the detailed arrangement of the quantization processor 3. A quantization process that converts color component data into ternary data will be exemplified below. However, 10 the present invention is not limited to ternary conversion, and the same applies to binary conversion, quaternary conversion, and the like.

As will be described later, the quantization 15 processor 3 executes a process for associating 8-bit input luminance data R, G, and B with quantized data Y, M, C, and K used in the print section 4.

The quantization processor 3 shown in Fig. 6 roughly comprises a pre-processing block, a CD method 20 block that implements color difference diffusion of an output pattern, and a block of various memories and their interfaces.

R, G, and B 8-bit luminance data that have undergone color conversion by the color converter 2 25 undergo processes such as edge emphasis, noise removal, resolution conversion, and the like as needed by a pre-processor 201, and are then input to a color space

conversion circuit 202.

The R, G, and B data are converted from the RGB color space into a CMY color space in the color space conversion circuit 202, and C, M, and Y data then  
5 undergo gamma conversion and the like required to convert into an image that reflects the characteristics of the print section 4 in a correction circuit 203. Note that the color space conversion, gamma conversion, and the like normally use conversion tables. It is  
10 advantageous to combine these conversion tables into one table in terms of implementation.

The C, M, and Y data output from the correction circuit 203 are added with color difference data of a previous line and those of a previous pixel (to be  
15 described later) in a color difference addition circuit 204.

Then, a combination pattern (output pattern) of inks which reproduce a color closest to the C, M, and Y data added with the color difference data on the color  
20 space is selected. This selection is made based on the CD method. It is preferable to calculate the distances between C, M, and Y data and output patterns and store them in a table (output color table) in advance, since the processing load can be reduced. Note that the  
25 distance calculations may be made on other color spaces (L\*a\*b\*, YCbCr, and the like).

Furthermore, in this embodiment, an upper bit

extraction circuit 205 extracts the upper 3 bits (total of 9 bits) of each of the C, M, and Y data (total of 24 bits) added with the color difference data, and input them to an output color table so as to further reduce the processing load.

That is, the 9-bit data (total) output from the upper bit extraction circuit 205 are input to a table memory 210 that stores an output color table via a RAM I/F 209, and C, M, Y, and K 1-bit data (output pattern) as print data corresponding to the 9-bit data (total) are output to a buffer circuit 211 via the RAM I/F 209.

When input data is converted into a smaller number of bits, errors (color differences) are generated inevitably, and color reproduction impairs due to the errors. Hence, the color differences are diffused to surrounding pixels using the CD method to avoid deterioration of color reproduction. Details of color difference diffusion will be described below.

## • Diffusion of Color Difference

A subtraction circuit 206 shown in Fig. 6 calculates the color differences between the density information of input data and that of the selected output pattern. As the density information of the output pattern, information obtained by predicting the density upon printing that pattern under a predetermined condition or information experimentally

obtained by measurement is used. The predicted or measured density information is stored in the table memory 210 as an output density table corresponding to the output color table, and density information  
5 corresponding to the selected output pattern is supplied to the subtraction circuit 206 via the RAM I/F 209.

A color difference distribution circuit 207 distributes the calculates color differences to  
10 respective pixels. That is, after the color differences are weighted, they are supplied to the color difference addition circuit 204 to be added to the next pixel, and are also supplied to a color difference integration circuit 208 to be diffused to  
15 pixels in the next line. The color difference integration circuit 208 calculates the sum of the color differences for each pixel corresponding to that in the next line, and stores the sum in a previous line color difference memory 213 via a RAM I/F 212.

20 Fig. 7 is a flow chart showing the process to be executed by the quantization processor 3.

Upon execution of a color copy or print process, the previous line color difference memory 213 and respective buffers in the color converter 2 are cleared  
25 (S101). A counter PIX which indicates the pixel to be processed is reset to zero (S102). After a document image is read (or after image data is input), image

data that have undergone various pre-processes are written in registers, which are assigned to R, G, and B data respectively, of the quantization processor 3 (S103). In case of this embodiment which converts the resolution from 300 dpi to 600 dpi, the number of accesses can be reduced and the resolution conversion efficiency can be improved when data for two pixels (R0, R1, G0, G1, B0, and B1) are input at the same time.

10           As a method of managing the process of the quantization processor 3, a method of counting clocks is available. A system counter starts to count clocks in response to data written by the CPU 11 as a trigger. When the processing timing is controlled on the basis of the count value of the system counter, an optimal processing circuit can be designed. Also, since processes (circuits) synchronized with clocks can be realized, timing verification and the like can be easily done.

20           For example, when data are written in the order of addresses R, G, and B in the process in step S103, the system counter starts to count clocks in response to data written at address B as a trigger. The system counter is cleared by software or in response to next data written at address R.

25           Interpolated pixels Rc0 and Rc1 are generated from pixels R0 and R1 (S104). Upon converting the



resolution from 300 dpi into 600 dpi, Rc0 is generated by bit-shifting the previous pixel and R0 and adding them. G and B pixels are similarly interpolated.

The color space conversion circuit 202 converts  
5 the resolution-converted R, G, and B data into C, M, and Y data (S105). Note that RGB  $\rightarrow$  CMY conversion can be independently executed for each color. If the processing performance has a margin against the processing load, R, G, and B data may be converted into  
10 data on another color space.

In order to convert into an image that reflects the characteristics of the print section 4, the correction circuit 203 applies gamma conversion and the like to the C, M, and Y data (S106).

15 With the above processes, data Ci, Mi, and Yi are generated.

Subsequently, color difference data CL of the previous line to be diffused to the pixel to be processed is read out from the previous line color  
20 difference memory 213 (S107). This color difference data CL will be described later. Subsequently, as shown in Fig. 5, the color difference addition circuit 204 adds the color difference data CL of the previous line and color difference data CP of a previous pixel  
25 (a pixel processed by the immediately preceding process) to the data Ci of the pixel to be processed (S108), and stores the sum  $C_i + CL + CP$  in a buffer of

the color difference addition circuit 204 as data ILPC.  
This buffer can store signed 11-bit data (-512 to  
+512). When ILPC exceeds 512, it is rounded to 512 to  
prevent the buffer from overflowing. Hence, the  
5 circuit scale of the color difference addition circuit  
204 can be reduced within the range that does not  
influence an image.

Similar processes are parallelly executed for the  
data Mi and Yi to store data ILPM and ILPY in the  
10 buffer. In this manner, data obtained by adding the  
color differences of the predetermined pixel in the  
previous line and previously processed pixel to data of  
the pixel to be processed can be obtained.

Address data ILPC', ILPM', and ILPY' are  
15 generated by extracting the upper 3 bits of each of the  
calculated data ILPC, ILPM, and ILPY (S109), and are  
input to an output color table shown in Fig. 9 to  
obtain an output pattern (Co, Mo, Yo, and Ko) closest  
to the color of input data on the color space. The  
20 obtained output pattern is stored in a register. Note  
that the output color table has the following  
configuration.

On the color space, distance L between colors  
defined by input data ILPC, ILPM, and ILPY, and colors  
25 Cp, Mp, and Yp indicated by color measurement data  
actually obtained by the print section 4 is calculated  
by:

$$L2 = (ILPC - Cp)^2 + (ILPM - Mp)^2 + (ILPY - Yp)^2 \dots(1)$$

The values Yp, Mp, and Cp are changed within a range shown in the output color table of Fig. 9 to find a combination of Cp, Mp, and Yp, which yields a minimum value L2. Then, an output color table which stores ternary data Ko, Co, Mo, and Yo corresponding to combinations of Cp, Mp, and Yp in correspondence with input data ILPC, ILPM, and ILPY is generated.

When the above calculation is made to obtain output color data of each pixel, the load on software and hardware becomes considerably heavy, and a long processing time is required. To avoid this problem, in this embodiment, the above calculation results are pre-stored in a table, thus quickly obtaining an output pattern (Ko, Co, Mo, and Yo) closest to the color of input data on the color space on the basis of the input data ILPC, ILPM, and ILPY that take the color differences into consideration.

Furthermore, when a plurality of output tables for the CD method corresponding to the characteristics and print method of a printer, the characteristics of print agents such as inks, toners, or the like, and the characteristics of print media are prepared on the table memory 210, an optimal output pattern (Ko, Co, Mo, and Yo) can be obtained quickly by using the CD method. Also, the console 15, an external personal

computer, or the like can input the print method, the characteristics of print agents and the characteristic of print media, and can change or designate the output color table to be used can be changed or designated, thus an image can be expressed using a plurality of dots or dots having different sizes. Note that the output color table inputs higher-order 3 bits each of ILPC, ILPM and ILPY as an input address, thus a memory size of the output color table is reduced and the table memory 210 can store a plurality of the output color table.

The input data ILPC, ILPM, and ILPY are respectively signed 11-bit data, as described above, but each of data Ko, Co, Mo, and Yo of the output pattern is several bits (2 bits in this embodiment). In this embodiment, the size of the output color table is reduced by obtaining output color data using the upper several bits of input data. That is, the output color table can be formed of 512 sets of table data using address data ILPC', ILPM', and ILPY' of the upper 3 bits (a total of 9 bits) of each of the input data ILPC, ILPM, and ILPY by removing their sign.

Next, the output pattern (Ko, Co, Mo, and Yo) is stored in the buffer circuit 211. This output pattern indicates a combination of output dots. Registers are respectively assigned to C, M, Y, and K data, and the processes in steps S103 to S110 are repeated until data

for eight pixels corresponding to the data bus size are stored (S111). After the substitution result data for eight pixels are stored in the buffer circuit 211, these data are read out (S112).

5           By reading out the substitution result data every eight pixels, the number of accesses from the CPU 11 is reduced. Furthermore, when the print section 4 processes data for each line, the load on a data rearrangement process by software can be reduced. When  
10 data for two pixels are output in the sub-scan direction, the output bits are separately written in independent registers, so that these data can be output to the print section 4 as data for different lines. If the CPU 11 has a transfer mode for each word  
15 (16 bits), two each registers may be assigned to each of C, M, Y, and K data, and the data may be read out after data for 16 pixels are stored.

          Upon completion of quantization of data for one line (S113), the line to be processed is incremented,  
20 and the flow returns to step S102. Upon completion of quantization for one page (S114), if the next page to be processed remains (S115), the flow returns to step S101; otherwise, the process ends.

## 25   • Diffusion of Color Difference

Fig. 10 is a flow chart showing an example of the process for diffusing color differences produced in the

quantization process.

A counter y indicating the vertical position of the pixel to be processed is reset to zero (S201), and a counter x indicating the horizontal position of the pixel to be processed is reset to zero (S202).

An output density table shown in Fig. 11 is looked up on the basis of the output pattern (Co, Mo, Yo, and Ko) obtained from the output color table, thus obtaining multi-valued data Cp, Mp, and Yp (S203).

Note that the data Cp, Mp, and Yp indicate the densities of a color printed on the basis of Co, Mo, Yo, and Ko, as described above.

The subtraction circuit 206 calculates color differences Ce, Me, and Ye between the input data Ci, Mi, and Yi, and output density data Cp, Mp, and Yp (S204).

$$Ce = Cp - Ci$$

$$Me = Mp - Mi \quad \dots(2)$$

$$Ye = Yp - Yi$$

The color difference distribution circuit 207 diffuses the color differences Ce, Me, and Ye to neighboring pixels. More specifically, as exemplified in Fig. 12, the color differences are diffused to four pixels, i.e., a right pixel of the pixel to be processed, and a lower left pixel, lower pixel, and lower right pixel in the next line. Diffusion to the right pixel is achieved by sending diffusion data to

the color difference addition circuit 204 (S205), and diffusion to the three pixels in the next line is achieved by sending diffusion data to the previous line color difference memory 213 (S206). Also, the

5 diffusion ratio is as follows:

$2/16 \cdot E$  to lower left pixel  $(x-1, y+1)$

$5/16 \cdot E$  to lower pixel  $(x, y+1)$

$1/16 \cdot E$  to lower right pixel  $(x+1, y+1)$

$8/16 \cdot E$  to right pixel  $(x+1, y)$

10 where  $E$  is the color difference of a pixel  $(x, y)$  to be processed.

By further distributing the remainder obtained by dividing the color difference by 16 to the right pixel  $(x+1, y)$ , the color difference information can be  
15 reflected without any losses.

Therefore, the color difference addition circuit 204 makes addition (S205) described by:

$$Y_i(x+1, y) = Y_i(x+1, y) + Y_e/2$$

$$M_i(x+1, y) = M_i(x+1, y) + M_e/2 \quad \dots(3)$$

20  $C_i(x+1, y) = C_i(x+1, y) + C_e/2$

Thus, color difference  $E(x, y)$  to be diffused to the pixel  $(x, y)$  to be processed is given by:

$$\begin{aligned} E(x, y) = & 1/16 \cdot E(x-1, y-1) + 5/16 \cdot E(x, y-1) \\ & + 2/16 \cdot E(x+1, y-1) + 8/16 \cdot E(x-1, y) \end{aligned}$$

25  $\dots(4)$

In the above equation, diffusion data for the previous line  $(y-1)$  is stored in the previous line

color difference memory 213. That is, as described above, the color difference integration circuit 208 stores, for a set of pixels of the previous lines, the sum of color differences given by:

$$\begin{aligned} \text{CL} = & 1/16 \cdot E(x-1, y-1) + 5/16 \cdot E(x, y-1) \\ & + 2/16 \cdot E(x+1, y-1) \quad \dots(5) \end{aligned}$$

In this way, the memory can be effectively used.

In order to reduce the number of accesses by utilizing a memory chip, the diffusion coefficients shown in Fig. 12 can be determined so that the color difference to be stored falls within 8 bits (-128 to +128). By broadening a region to which the color difference is diffused, more faithful color reproduction can be obtained. Since the color difference is diffused to unprocessed pixels, if the processing direction is alternately switched between the right and left directions, the color difference can be prevented from being diffused in only a specific direction, thus suppressing deterioration of image quality such as moiré and the like.

The counter  $x$  is incremented (S207), and it is checked if the process for one line is complete (S208). Upon completion of the process for one, the counter  $y$  is incremented (S209), and it is checked if the process for one page is complete (S210). If the process for one page is not complete yet, the flow returns to step S203.



• Determination of Output Color Table

Combinations of output colors used by the system are determined first. There are eight combinations of  
5 output colors when output data for two pixels in the sub-scan direction are generated in response to one pixel input, even when both the pixels are formed of a combination of identical color dots.

A case wherein two dots have different colors  
10 will be explained with reference to Figs. 13 and 14.

If Y and K dots are not used, when  $C = 0$  and  $M = 0$ , neither of dots are printed to form white (print medium color). When  $C = 1$  and  $M = 0$ , only a C dot is printed; when  $C = 2$  and  $M = 0$ , two C dots are printed;  
15 and when  $C = 2$  and  $M = 2$ , two C dots and two M dots are printed. In this way, color spaces corresponding in number to combinations of Y and C, M and Y, and respective colors and K can be expressed (see Fig. 14).

Arithmetically,  $3^4 = 81$  different combinations  
20 are available. However, colors such as  $(C, M, Y, K) = (1, 1, 1, 2)$  and the like, which can be satisfactorily reproduced by other combinations, and combinations which cannot be used due to limitations such as ink dot size limitations and the like are excluded. The  
25 remaining combinations of output patterns are printed as color patches on an actual print medium, and the printed color patches are measured. That is, the color

spaces of data obtained by reading the color patches by the scanning unit 14 are converted by the color space conversion circuit 202. The obtained colorimetric data are stored in a table shown in Fig. 15 as data Cp, Mp, and Yp obtained from the patches.

Using equation (1) above, distance L between ILP, and Cp, Mp, and Yp is calculated, and the table shown in Fig. 11 is generated from Cp, Mp, and Yp that can minimize distance L, i.e., C, M, Y, and K data used to generate the color patches.

As described above, the CD method of this embodiment substitutes all input colors by combinations of output colors (combinations of color dots) closest to the input colors, and diffuses color differences generated at that time to surrounding pixels. Therefore, the way input colors correspond to combinations of output colors becomes a basis upon improving the color reproducibility using the CD method.

As described above, since C, M, Y, and K data must be obtained for ILPC, ILPM, and ILPY data, and Yp, Mp, and Cp data must be obtained for C, M, Y, and K data, a table shown in Fig. 16 obtained by combining the tables shown in Figs. 9 to 11 is stored in the table memory 210.

Note that the correspondences between ILPC, ILPM, and ILPY, and C, M, Y, and K, and those between C, M,

Y, and K, and Yp, Mp, and Cp vary depending on inks or print media to be used even when the print section 4 remains the same. Therefore, when the table shown in Fig. 16 is switched in correspondence with inks, print media, and the like supported by the print section 4, high color reproducibility can be obtained. In this case, a table may be read out from the ROM or the like for each copy or each setup change without storing a plurality of tables in a RAM such as the table memory 210 or the like, and may be stored in the table memory 210. Conversely, an average table common to inks, print media, and the like supported by the print section 4 may be stored in the table memory 210, thus simplifying the table.

15

#### [Generation of Table]

The design method of the table of the quantization processor 3 and 3D-LUT 31 will be briefly described below.

20       The characteristics of the quantization processor 3 are specified by print media (paper types and the like) and print materials (characteristics of inks and the like). Therefore, the table of the quantization processor 3 is designed to be able to diffuse color differences on a space expressed by primary colors of respective inks in place of a simple density space, so as to express maximum and sufficient gray levels. That

25

is, the table is defined to maximize the spaces of colors that can be expressed with respect to values input to the quantization processor 3.

On the other hand, the 3D-LUT 31 defines which  
5 RGB input values correspond to which colors of output data. For example, when a plurality of tables such as a table that increases the saturation of a specific hue, a table that increases the contrast of a middle luminance, and the like are prepared as the 3D-LUT,  
10 images having various features (color appearances) can be obtained.

As described above, according to this embodiment, preferred color conversion that matches respective hue, saturation, and lightness levels can be made with high  
15 precision, and color separation which is optimal to an output device and respective print modes of that device, and also to print media can be processed at high speed.

Furthermore, color separation that flexibly cope  
20 with user's needs (e.g., "quick", "fine") and/or print media to be used, and preferred color conversion can be independently made. Processes corresponding to huge combinations of the characteristics of input/output devices, user's color favors, modes of user's choice,  
25 types of media, and the like can be efficiently done, and preferred color reproduction can be achieved.

<Other Embodiment>

The present invention can be applied to a system constituted by a plurality of devices (e.g., host computer, interface, reader, printer) or to an apparatus comprising a single device (e.g., copying  
5 machine, facsimile machine).

Further, the object of the present invention can be also achieved by providing a storage medium storing program codes for performing the aforesaid processes to a system or an apparatus, reading the program codes  
10 with a computer (e.g., CPU, MPU) of the system or apparatus from the storage medium, then executing the program.

In this case, the program codes read from the storage medium realize the functions according to the  
15 embodiment, and the storage medium storing the program codes constitutes the invention.

Further, the storage medium, such as a floppy disk, a hard disk, an optical disk, a magneto-optical disk, CD-ROM, CD-R, a magnetic tape, a non-volatile  
20 type memory card, and ROM can be used for providing the program codes.

Furthermore, besides aforesaid functions according to the above embodiment are realized by executing the program codes which are read by a  
25 computer, the present invention includes a case where an OS (operating system) or the like working on the computer performs a part or entire processes in

accordance with designations of the program codes and realizes functions according to the above embodiment.

Furthermore, the present invention also includes a case where, after the program codes read from the  
5 storage medium are written in a function expansion card which is inserted into the computer or in a memory provided in a function expansion unit which is connected to the computer, CPU or the like contained in the function expansion card or unit performs a part or  
10 entire process in accordance with designations of the program codes and realizes functions of the above embodiment.

The present invention is not limited to the above-described embodiments, and various changes and  
15 modifications can be made within the spirit and scope of the present invention. Therefore, in order to apprise the public of the scope of the present invention, the following claims are made.